10/593660
IAP9/Rec'd PCT/PTO 20 SEP 2006

Translation from French of answers to the International Search Report international application No. PCT/FR2005/000659

References :

International application No.	PCT/FR2005/000659
International filing date	March 18, 2005
Priority date	March 23, 2004
Applicant	PIATON, Alain Nicolas

10/593660
IAP9/Rec'd PCT/PTO 20 SEP 2006

Answer to the International Search Report
Application **PCT/FR2005000659** of March 18th, 2005

Preamble

At the time of the filing of the patent application on which the priority of the PCT cited in reference is based, search engines on e-mail messages and documents contained in PCs, called Desktop Search software, were practically unknown to the general public.

Since then, tools such as Google Desktop, Windows Desktop Search, Copernic, etc., which are used by hundreds of millions users across the world, allow for a better knowledge of the architecture of these different kind of tools.

The process according to the invention describes an original implementation to build a Desktop Search, with an essential particularity, namely that it works without an index table.

Indeed, all other search engines work with an index table (also called thesaurus or dictionary or inverse list or inverse index table), according to a device that has been known for a long time by all librarians, namely:

- One begins by constituting a thesaurus (or dictionary) with:
  - Concepts such as "finance" (or financier, financing, financially, etc.),
  - Names of persons, companies, places, cities, countries, etc...
- Then, one scans each document
  - Skipping meaningless words, that is articles (the, of, ...), conjunctions (and but or so...), adjectives such as big, small, etc...
  - And by keeping the words contained in the thesaurus, putting them in a list.
- Finally, one only has to modify the table of contents (that contains all the thesaurus words), and for each word that appears in the list, add a reference to this new document.

In this way, when one wants to know if a given word appears in the document, one just has to consult this list, generally sorted by alphabetical order, to know how many times the word occurs, and at what location. This is the device used to cite bibliographic references that one finds at the end of scientific literature.

This technique is known since antiquity (the library of Alexandria!), and one the index table is made, to work, the search engine requires only 2 permanent elements, namely the original information, and the index table, that contains part of the words contained in the original document, and that are stored in a very remote way from the original information.

The process according to the invention is of a different nature.
It is a little as if, to find the places where one mentions an author, at each times one would read the totality of the documents, which apparently is a bad solution, except if the quantity of documents is low, and if the scan can be done very rapidly.

This search engine type may work without resorting to « artificial intelligence » since in its principle it just looks for a character string in another character string.

The table mentioned in the process according to the invention, is equivalent to the totality of the original documents, in **plain text** form, (as in the case of an e-mail message displayed in plain text format, that contains the same text as the HTML message, but without typesetting, or color, etc.)

To be in conformity with the process as described below, the claims number 1 and 23, which were too general, have been modified.

Comments

About the 3 documents cited in the International Search Report:

X – EP 0 886 227 A (Digital Equipement)
X – US 6721 748 81 (Knight Thimothy)
X – WO 02/065316 (OTG Software)

X – US 6721 748 81 (Knight Thimothy)

- This process making use, as the previous one, to **artificial intelligence**, proposes a solution to establish automatically correlations between different documents, with an additional difficulty, namely that each document or answer that one finds in a forum must be analysed in a context, which is very difficult.
As for the Schliter Théo process, this system not only takes the original plain text, but is based on a very complex index tables analysis and management, tables that are not of any use to display results in a preview window.

There is no mention anywhere of memorizing the totality of the « meaningful » information (**plain text**) equivalent to the original information, or of sequentially scanning it.

X – EP 0 886 227 A (Digital Equipement)

This process describes a search engine in e-mail messages using an inverse index table, and may work **without** "artificial intelligence", which is useful for a "desktop search".

The term «full-text index» (col 2 line 12, col 6 line 24, col 8 line 30...) may lend to confusion, because it means that all words that appear in the original documents will appear in the generated inverse index table, including words that do not have strong meaning like the word « is » which appears in « this is me » et « hello me » -see fig 5;
(as if the librarian would put in his thesaurus the articles « the, of.... »).

The interest of this process is to generate and maintain in a judicious way this table, and the links in this table with the references to the original documents (e-mails).
L'intérêt de ce procédé est de générer et maintenir de façon judicieuse cette table, et la liaison de cette table avec les références des documents de départ (mails). On the other hand, there is no mention anywhere of memorizing the totality of the « meaningful » information (**plain text**) equivalent to the original information, or of sequentially scanning it.

X – WO 02/065316 (OTG Software)

This process is the only one mentionning « **plain text** » ;
It is based entirely on the concept of « **message tag** » see the first line of [0011]:
« then computes an unique identifier or Message Tag... »,
And it appears explicitly in claim 1
« computing a Message Tag from at least a portion or a plurality of message property... ",
Then in the other pivot claims 8, 15, 21, 30 and 37.

The aim of this Message Tag, is to obtain an absolutely unique identifier, the shortest possible one even there are tens of millions of messages sent and received, as it is the case with the 35000 employees of a big French bank who exchange 70 millions messages per year, that is to say a third of a billion messages in 5 years. This identifier is a search criterion for the relational database (Oracle or other)
"the index file comprises a list of message tags corresponding to all messages stored in ..." (see [0036]).

Except for some keywords, such as the date, the sender e-mail address, the subject and a few metadata, a relational database (see [0044]) is not used to find a message as all the search engines available on the market;

As to the idea of storing also the **plain text** of the message body, or the plain text of the attachments, in a relational database this is not the solution adopted by the desktop search software available on the market, because it seems difficult to do joins on 300 million plain text pieces.

Anyway, even if it is about the same **plain text** in the 2 cases, the fact to store it piece by piece in a database is completely different from the constitution of a single table where all the texts are stored sequentially, then scanned sequentially.